A systematic literature review of green software metrics

Lago, P.; Gu, Q.; Bozzelli, P.

2014

A systematic literature review on green software metrics

Paolo Bozzelli*
Department of Computer Science
VU University Amsterdam
The Netherlands
p.bozzelli@student.vu.nl

Qing Gu†
Department of Computer Science
VU University Amsterdam
The Netherlands
q.gu@vu.nl

Patricia Lago‡
Department of Computer Science
VU University Amsterdam
The Netherlands
p.lago@vu.nl

ABSTRACT

Green IT is getting increasing attention in software engineering research. Nevertheless energy efficiency studies have mostly focused on the hardware side of IT, the software role still requires deepening in terms of methods and techniques. Furthermore, it is necessary to understand how to assess the software "greenness" for stimulating the energy efficiency awareness, since early phases of the software lifecycle.
The main goal of this study is to describe and to classify metrics related to software "greenness" present in the software engineering literature. Furthermore, this study analyzes the evolution of those metrics, in terms of type, context, and evaluation methods.
To achieve this goal, a systematic literature review has been performed surveying the metrics claimed in the last decade. After examined 960 publications, we selected 23 of them as primary studies, from which we isolated extracting 96 different green metrics. Therefore, we analyzed search results in order to show what is the trend of research about green software metrics, how metrics perform measurement on resources, and what type of metrics are more appealing for defined contexts.

1. INTRODUCTION

Reducing energy consumption and carbon footprint, in order to achieve high levels of sustainability, are some of the challenges that the IT community is pursuing to deal with environmental issues generated by IT systems. Indeed, research community is performing a huge effort to detect the main reasons of energy consumption and a number of studies have been published in this specific theme. Moreover, a model to identify energy consumption sources and measure the utilization of energy is necessary, in order to generate a plan to reduce energy consumption. However, most of the studies focus on the hardware perspective of energy consumption measurement.
The aim of this work is to identify energy consumption metrics related to the software perspective and to classify these metrics in order to define the utilization purpose, the kind of measurement results, and the environment in which they are used. In order to achieve this goal, we perform a systematic review about green software metrics in the software engineering literature.
First, we define our research questions and, hence, our search strategy, identifying a set of suitable keywords to execute the search on a predefined set of data sources. Then, we select a number of primary studies that are compliant to a predefined set of inclusion and exclusion criteria. Furthermore, we extract metrics from these primary studies and classify them with respect a predefined set of subjects. Finally, we show the results of our work, and then we analyze these results in order to show which is the trend of that research about these metrics has been focusing on, how these metrics perform measures on related resources, and which metrics are more appealing with respect to their environment.
We organize the content of this paper as follows: in Section 2, we describe in detail the research method we used, eliciting research question and defining the review protocol; in Section 3, we show the results of our search and the classification of found metrics; in Section 4, we analyze the results in order to investigate metric properties, common features and utilization; in Section 5, we state our conclusions and discuss about possible applications and reuse of our work.

2. RESEARCH METHOD

In this section, we describe in detail the research method used for this systematic review. We firstly define the research questions, which define the aim of our research, and then we describe which protocol has been performed to search and collect studies about green software metrics.

2.1 Research questions

The aim of this work is to find out how many green software metrics are defined in the software engineering literature, and then classify them in clusters for later reuse. Our corresponding research questions are:

Q1 What green metrics have been proposed in the Software Engineering literature?

*Paolo is a MSc student who carried out the main body of the work.
†Dr. Gu was the daily supervisor of Paolo and assisted Paolo in accomplishing the work.
‡Dr. Lago was the supervisor of Paolo and provided scientific advises during the work

Q2 How green metrics can be classified?

2.2 Review protocol

The review protocol for a systematic literature review is a set of tasks that have to be performed in order to answer the research questions listed in the Section 2.1, and to achieve complete and consistent results.
Our review protocol is made of five components: the *data sources* that define which scientific database have to be queried in order to find relevant studies; the *search strategy*, which states how those data sources have to be queried; the *study selection*, which defines criteria to select studies extracted during the search strategy execution; the *data extraction*, which describes how relevant data about green metrics are extracted from each of selected primary studies; and the *data synthesis*, which helps to classify the extracted results.
Each component is described in the following sections.

2.2.1 Data sources

To execute the search on the software engineering literature, we select the following six electronic libraries and perform the search on each of them.

- IEEE Explore[1]
- ACM Digital Library[2]
- ISI Web of Knowledge[3]
- SpringerLink[4]
- ScienceDirect[5]
- Wiley InterScience[6]

2.2.2 Search strategy

In order to define a search strategy, we start from the research questions and focus on which terms define the studies that we want to examine.

Keywords definition. Both Question 1 and 2 are well-defined, so that we can extract key terms like *green*, *metrics* and *software engineering*, since they are strictly related to the goal of this study.
Although, these key terms are not specific enough, and the execution of a query containing only these terms may be not expressive enough. For these reasons, the set of key terms is enhanced with synonyms or terms related to the same topic (i.e., "sustainable" is a synonym of "green", "evaluation" is a term related to the "metrics" topic), as suggested by Brereton et al. in [5], and by Kitchenham in [20].
"Green"-related keywords — Since we dealt with green-related topic in previous academic experiences, we rely on the literature provided for those works to find an expressive set of keywords.
Kurp in [21] gives a definition of green computing movement as "a multi-faceted, global effort to reduce energy consumption and promote sustainability". Furthermore, in [30] de Rijk states that the benefits of green IT may affect the "Brand Image, especially where customers (or their customers) care about green issues, environmental sustainability can be used as a competitive differentiator". Moreover, Kurp in [29] states that the goal of his study is "the development of context-aware and sustainable information systems". Then, we can state that research is focusing on development of sustainable system and sustainability can be both used to create green awareness to the customers and encouraged to achieve energy consumption goals. For these reasons, *sustainability* and *sustainable* are included as keywords.
In [4] Arnaud and MacLean proposed that WPIE – Working Party on the Information Economy – has to "focus on a new issue that is emerging in response to what has clearly become the dominant environmental concern of our time", stating that "in general this work has either not included issues of environmental sustainability such as climate change, or not approached them with the same kind of rigour that has been applied to the analysis of economic and social issues." This statements suggest us both to focus on environmental aspects of IT systems development and to search for studies that deal with environmental impact of those systems. Hence, *environmental* is an eligible keyword for the search query string.
In [29], Pernici et al. focus on "the point that energy efficiency should be given a very relevant role in Information Systems design." Furthermore, in [23] Lefèvre and Pierson state that "the use of ICT to improve energy efficiency and reduce costs is the subject of a number of papers in this special theme". According to deRijk, "software impacts hardware energy efficiency" [10]. Since energy efficiency is largely discussed in the literature and software influences directly the system energy efficiency, we add *energy efficiency* and *energy efficient* to the keywords set.
Several projects and studies are presented in order to achieve energy awareness: Meijer et al. present a project to minimize carbon dioxide emission that leads to a "significant step towards energy-aware, emission-free computing" [25]; moreover, Öhman presented a study about design for energy awareness [28]; last but not least, Kahn et al. introduce a work about energy-aware storage benchmarks [15]. Since there is a relevan number of studies that are proposing solutions and techniques to achieve energy awawreness, we choose *energy-aware* as a keyword for the search strategy.
In [30], Potter states that the "faster you get to greener using external IT services [...] the quicker an IT organization or enterprise can enjoy the benefits of environmental friendliness or compliance"; furthermore, he states that "optimization endeavors now often have an environmental friendly benefit". Measuring environmental friendly benefits should be interesting since it allows to quantify the real value of those benefits. For this reason, we select *environmental friendly* as a keyword.
"Metrics"-related keywords — Regarding to metrics related keywords, we search "metrics" synonyms within scientific dictionaries, such as Oxford Dictionary. Then, we search for those synonyms within the studies we mentioned above, in order to scope the set of keywords to terms that are related to the context of our work.
In [29], Pernici et al. mention a study by Williams [36],

[1] http://ieeexplore.ieee.org/
[2] http://dl.acm.org/
[3] http://apps.webofknowledge.com/
[4] http://www.springerlink.com/
[5] http://www.sciencedirect.com/
[6] http://onlinelibrary.wiley.com/

which discusses the necessity to create an "ecosystem map, introducing the concept, to be developed, of Key green performance indicators". Furthermore, in [12], Erdmann et al. chose a set of indicators to evaluate the environmental impact within selected economic sectors and ICT applications. Hence, we pick *indicator* as a keyword in the "metrics"-related set.
In [22], Lago and Jansen define green metrics as a tool to measure the actual carbon footprint of SBAs. This makes *measure* an eligible keyword for our search strategy.
In [29], Pernici et al. deal with the data redudancy problem as an example of energy saving from the software perspective; they propose to deduplicate redundant data and considering only useful data, stating that data relevance evaluation is still an open issue in the literature. Since we expect to find relevant results in the literature, we include *evaluation* as a keyword for our search strategy.
In some studies we previously dealt with, we found some definition of labelling methods with the aim of showing carbon emissions or carbon footprint of IT systems. In [10], deRijk cite the PAS 2050 standard[7], a specification that provides a method for assessing the life cycle greenhouse gas emissions of goods and services, and Logica's EMERALD[8], that measures the carbon equivalent of the six most harmful green house gases that contribute to climate change. Since labelling could be appropriate way to classify the greenness of a software, we include *labels* as a keyword in the "metrics"-related set.
In [30], Potter states that "IT organizations need to identify baselines and KPIs to measure progress against green IT and corporate green goals". Evaluating performance with respect to energy consumption should allow IT organization to quantify the effects and the benefits of the application of green policies. For this reason, *KPI* – key performance indicator – is then included as a keyword for our search strategy.
"Software engineering"-related keywords — We want to find studies about software-related energy consumption.
As stated by Lago and Jansen in [22], "few initiatives, though, measure how do software systems actually use these devices, with the goal of optimizing consumption of devices and computing resources".
In detail, in [29] Pernici et al. adopt a service-oriented approach for the development context-aware and sustainable information systems where energy consumption reduction is considered at the technological level; furthermore, Lago and Jansen in [22] propose a service-oriented approach to address three main problem areas to realize green service-based applications. For those reasons, we choose *software* and *service* as keywords related to software engineering.

Query string definition. Then, a query string is created using Boolean "OR" operator among related terms, and the "AND" operator among the three sets of keywords. Hence, the resulting query string is the following:

[QS1] *(green OR environmental OR sustainable OR sustainability OR "environmental friendly" OR "energy efficiency" OR "energy efficient" OR "energy-aware" OR "green computing" OR "green IT") AND (metrics OR measures OR indicators OR KPI OR evaluation OR labels) AND ("software engineering" OR "software" OR "service")*

To obtain meaningful results that make our study selection feasible, we first performed a number of search attempts. We realized that the set of keywords is too vast and the set of results makes the study selection unfeasible, due to the high number of studies resulting from the search on each data source. For example, performing a search by means of QS1 on IEEEXplore results in 452 studies. If we perform the search getting numbers like this on every data source listed in Section 2.2.1, the final set of result – ignoring duplicates – would result in about three thousand studies. Obviously, this makes the study selection unfeasible and difficult to perform.
For this reason, we need to scope our research on a smaller set of keywords. First of all, we take a look to the "green"-related terms. Since we already have a "green" term, we decide to remove similar terms like "green computing" and "green IT"; moreover, "computing" and "IT" are terms more related to the "software engineering" class. Then, we perform the search with the new set of keywords. However, the number of results is still the same.
To refine the search query and to make it meaningful and helpful for our aim, we decide to focus only on one of similar terms, e.g. "sustainability" and "sustainable". Since we intend "green" as a characteristic of the metrics we are looking for, we decide to pick only adjectives. This leads to crop out terms like "sustainability" and, hence, to keep "sustainable". Same rationale is used for "energy efficiency" and "energy efficient", and for "environmental" and "environmental friendly". So, we perform another search attempt and we figure out that the results are more scoped but still in a number that makes the study selection difficult to be performed (329 studies).
Further reductions of the "green"-related terms can make the search query string meaningless from the "green" perspective. So we focus on the "software engineering"-related terms. Terms like "software" and "service" can be correct but poorly descriptive and too vague. As we previously said, we add "computing" term to this set of keywords, in order to have "green computing metrics" as a possible query keyword combination. Since we are interested in techniques and methods for developing green software, we choose to scope the search with the "software development" term. Then, we attempt to see if the set of results has a reasonable number of studies: 109 results is an acceptable number of studies in order to perform the study selection properly.
Hence, the final set of keywords that will be used to perform the search on each data source is the one shown in Table 1.

Table 1: Keywords and related terms.

green	metrics	software engineering
environmental	measure	software development
sustainable	indicator	computing
energy efficient	KPI	
	labels	
	evaluation	

Furthermore, the query string used to perform the search is

[7]http://www.bsigroup.com/en/Standards-and-Publications/How-we-can-help-you/Professional-Standards-Service/PAS-2050/PAS-2050/
[8]http://www.logica.com/we-are-logica/media-centre/factsheets/emerald/

the following:

[QS2] *(green OR environmental OR sustainable OR "energy efficient") AND (metrics OR measure OR indicator OR KPI OR evaluation OR labels) AND ("software engineering" OR "software development" OR computing)*

Further search filters. Once the search query string is defined, two further details are added to the search strategy: the *range of time*, in which examined and selected studies are published, and the *document section* – such as title, abstract or full text – where to apply the search strategy.
We choose to scope the range of time of our research within 2000 and 2012. This is because our knowledge is based on the previously mentioned literature. The studies and the articles listed in this literature are published within 2004 and 2010; then, to get a relevant set of studies, we decide to focus on the studies published in the last decade.
Moreover, we decide to perform our search strategy on abstracts sections. Searching only in title section may drastically reduce the set of results, excluding relevant studies that do not contain any defined keywords. Searching in the full text may exponentially increase the number of found publications, including studies that may be out of the scope of this research.

2.2.3 Study selection

After the search strategy is executed, the study selection is performed on the resulting set of studies.
The selection is performed with respect to defined criteria, listed in Table 2.
These criteria focus on the quality of selected studies, in terms of source and language, and on the topic of this review, in terms of relevance and completeness.
In detail, inclusion criterion I3 and I4 are both related to the quality of the current study: the former filters the studies that focus on scientific contents, the latter includes only studies written in English language.
Moreover, inclusion criterion I1 and I2 are both related to the topic of our review. Inclusion criterion I1 limits this review to those studies that are about green software engineering. We want to focus only on studies that propose methods, techniques and approaches to design and develop green software. Inclusion criterion I2 selects studies that provide a complete and detailed description of a metric.
By contrast, E1 excludes studies that cope with different topics. E2.1 excludes those studies that propose metrics related to energy efficiency but from the hardware perspective, or that do not provide metrics at all. E2.2 excludes those studies that do not provide a detailed description of the metrics.
A study is selected if it fulfills ***all*** the inclusion criteria; otherwise, it is discarded if it fulfills ***any*** exclusion criteria.
Furthermore, the study selection is performed following a precise assessment order: for example, if the current study fulfills E4, it is discarded and no other criteria is examined; if it fulfills I4, but it fulfills E3, it is discarded and no other criteria is examined, and so on. The assessment order is the following:

$$E4 > E3 > E1 > E2.1 > E2.2$$

This assessment order is helpful for both study selection execution and quantitative analysis.
We rank criteria with respect to their easiness of assessment. For example, we first assess the exclusion criterion E4, since we can easily recognize if a study is whether written in English or not.
If the study fulfills this criterion, we can discard it right away, speeding up considerably the selection process. For example, the E2.2 criterion requires a "full-text" reading, whilst to assess the E3 criterion just requires a "title" reading.
Furthermore, following the assessment order, it is possible to count how many studies are excluded because of a certain criterion, since only the first fulfilled criterion is marked within the spreadsheet.

2.2.4 Cross-references check (CRC)

After reading and analyzing the studies, we apply the "snowballing" search method – as described by Greenhalgh and Peacock in [14] – to track all the references contained in the *References* section from each selected primary studies. Then, we perform both data collection and study selection on this new set of studies in the same way as they are described in the Section 2.2.3.

2.2.5 Data extraction

After performing study selection on both search strategy execution results and CRC results, we are able to start the data extraction on the primary studies. The goal of this stage is to collect all the metrics – and related information – mentioned within the primary studies.
Hence, we gather all the extracted information within a spreadsheet that contains the following fields:

Name – the name of the current study;
Year – the publication year of the current study;
Metric Name – the name of the current metric;
Metric Common Name – the name of metrics that have similar features and, hence, they can be counted as a single metric.
Metric Description – brief textual description of the metric;
Metric Calculation – the formula used to calculate the metric (if any);
Metric Unit – the measurement unit of the metric (i.e. kWh, seconds, etc.);
Measured Resource – the resource to be measured by means of the current metric (i.e. data centre, memory, etc.);
Software-related – whether the metric is related to software energy consumption or not;
Metric Type (Extracted Data + Rationale) – textual description – extracted from the study – of the kind of results generated by the current metric; optionally, a rationale to motivate the related type is provided;
Metris Type – name that identifies the type of the current metric;
Metric Context (Extracted Data + Rationale) – text – extracted from the study – describing the enviroment in which the current metric is involved; optionally, a rationale to motivate the related context is provided;
Metric Context – a name describing the context of the current metric;

Table 2: Inclusion and Exclusion Criteria.

I1 *A study is related to green software engineering* Motivation: The study proposes methods, techniques or approaches to design and develop green software. This would include studies about e.g. design notations for green software, evaluation of green software, reusable practices for green or sustainable software development.	**E1** *A study is not directly presenting any method, approach or technique to be used for green software engineering.* Rationale: Green software engineering is mentioned, but the work is about something else than green software development or green software systems/components. For instance, a paper defining a green version of CMMI would be included. Also a paper describing a set of metrics to measure energy efficiency of a software application would be included as such metrics would be useful in software engineering even if the paper does not discuss e.g. how they would fit in software engineering activities. However, a paper describing a set of metrics to measure energy efficiency of hardware devices would be excluded.
I2 *A study proposes a documented set of green-related metrics* Motivation: The study provides a set of that have a well-defined and unambiguous supporting description. For instance, a study may discuss about metrics designed for power consumption and energy efficiency, and provides a detailed description of those metrics.	**E2.1** *study is about green metrics that are not concerning software* Rationale: metrics may be related to a large set of topics and fields. Since we used "metrics" as a search keyword, it is possible to retrieve studies not related to methods or approaches concerning development and design of green software. E.g., a paper describing a set of metrics to measure energy efficiency of hardware devices would be excluded. **E2.2** *A study does not provide a description of metrics* Rationale: if there is a lack of information regarding metrics, and if no description is provided, it is impossible to define properly the whole set of green metrics. In addition, in case of similar metrics comparison, it would be impossible to verify analogies. For instance, a paper that mentions metrics or related quality factors without defining what to measure is excluded.
I3 *A study is in the form of a scientific paper* Motivation: the study focuses on scientific contents, in order to guarantee an good level of quality. For example, a study may be a journal part of a conference, and it may respect the standard publication templates (i.e. it contains abstract, introduction, description of the problem, proposed solutions, related work and references).	**E3** *A study is not in form of a scientific paper* Rationale: lack of scientific contents and rigorous methods can lead to a low-quality outcome. To meet our quality goals, non-scientific contents have to be ignored. For example, article from magazine, collection of abstracts, oral presentations, or reports can't be selected.
I4 *A study is written in English* Motivation: publications in Computer Science field are required to be written in English to be submitted.	**E4** *A study is not written in English* Rationale: languages such French, German, Spanish or similar do not fit with submission policies for publications in Computer Science. For instance, if a study is written in German, it is discarded.

Metric Purpose – whether the metric is designed either for measurement or estimation (or both).

2.2.6 Data synthesis

In this section, we explain how we classify the metrics, showing how we fill the spreadsheet we introduced in the previous section.
The field **Name** is simply filled with the title of the current study and the field **Year** is filled with its publication year.
The field **Metric Name** is filled with the name of the metric given within the study, whether it is a descriptive name (e.g. Availability) or an abbreviation (e.g. $E_{coord,i}$).
The field **Metric Common Name** may be filled in two different ways: if the study provides only an abbreviation as the metric name, we give a descriptive name; if more metrics are designed for the same purpose – e.g they are identically descripted, measure the same resource, and calculates values in the same way –, we assign the same Common Name to each similar metrics. The field **Metric Description** is filled with the textual description provided by the study claiming the metric.
The field **Metric Calculation** is filled with the eventual calculation formula provided by the study which the metric belongs to.
The field **Metric Unit** is filled with the measurement unit which with the metric is proposed.
The field **Measured Resource** is filled with the name of the resource that the metric uses to perform measurement. For example, if the metric evaluates the energy consumption generated by writing and reading operations performed on the RAM, the Measured Resource should be "Memory".
The field **Software-related** is filled with "Yes" if the current metric is designed for measuring the energy consumption of the software or for measuring aspects that are strictly related to software energy consumption. For example, if a metric evaluates the revenue earned by saving energy related to software services execution or measures the power consumption of an application, it is classified as Software-related. Otherwise, if the current metric is not related to energy consumption generated by software – e.g. the pollution generated by IT organization employees due to their transportation – this field is filled with "No".
The field **Metric Type (Extracted Data + Rationale)** is filled with a textual fragment from the study that recalls the kind of results generated by the current metric. If this text is missing, we fill this field providing a rationale that explains why the current metric is classified with a certain Metric Type. In addition, if the extracted text does not suffice, we fill this field both including the extracted text and providing a motivating rationale.
The field Metric Type is filled with a descriptive name that indicates the kind of results generated by the measurement perfomed with the current metric. For example, if a metric is designed to estimate the energy consumption of a system, the Metric Type should be "Energy".
The field **Metric Context (Extracted Data + Rationale)** is filled with a textual portion from the study that recalls the environment in which the metric performs measurements. If this text is missing, we fill this field providing a rationale that explains why the current metric is classified with a certain Metric Context. In addition, if the extracted text does not suffice, we fill this field both including the extracted text and providing a motivating rationale.
The field **Metric Context** is filled with a descriptive name that indicates the environment in which the metric performs measurements. For example, if a metric is designed to measure the energy efficiency of a service center and it is included in a set of proposed metrics that perform measurements on other aspects of a service center, the Metric Context should be "Service Center".
The field **Metrics Purpose** may be filled with three different entries: "Measurement", if the metric is designed to perfom real measurements or to calculate a value starting from one or more measurements; "Estimation", if the metric is designed to perform a calculation based on theoretical prediction or approximation of values; "Measurement / Estimation", if the metric is designed to perform an estimation starting from values generated by real measurements.

3. RESULTS

In the previous sections we explained the protocol we followed to identify the relevant primary studies and to extract relevant data about green software metrics.
In this section we show the results of this data extraction, focusing on the dimensions with which we classified the metrics.
This section is organized as follows: Section 3.1 shows selected primary studies resulting from strategy execution, study selection and cross-reference check stage; Section 3.2 shows all the found metrics, whether designed for measuring software energy consumption (namely *Software-related metrics*, SR metrics, hereafter) or not (namely *Non-software-related metric*, NSR metrics, hereafter); Section 3.3 shows all the types with which SR metrics have been classified, providing a detailed description of every type and defining the related measurement unit used to express measurement results; Section 3.4 describes in detail each context to which the SR metrics belong; Section 3.5 shows which resources have been measured by SR metrics; Section 3.6 shows how many SR metrics have been proposed to perform measurement or estimation of software energy consumption.

3.1 Selected primary studies

In this section, we show which primary studies we selected, illustrating the results generated by search strategy execution, study selection and CRC.

3.1.1 Search strategy execution results

As we stated in Section 2.2.1, we execute the search strategy on six different data sources.
We gather search results in two ways. First, we extracted search results as bibliography in BibTeX format, so that we get a final collection of bibliographies for each data source that has been queried; then, we used JabRef[9] to merge those set of bibliography in a unique *.bib* file, to detect and to remove duplicates.
Moreover, we used a spreadsheet in order to perform a quantitative analysis on search strategy execution results. This spreadsheet contains:

Data source – the name of the current data source;
Limitations – constraints that eventually are imposed by the data source search engine, (i.e. search query string length);

[9]http://jabref.sourgeforge.net/

Search query – the query string interpreted by the current search engine; it may be different in terms of syntax, but it is consistent in terms of semantic;
Further search filters – other search filters used in order to refine the search results set;
Results – the number of resulting articles;
Search URI – the (short) URI related to the search performed on the current data source;
.bib file – the .bib file of the search strategy execution results related to the current data source.

Data source	Results
IEEEXplore	107
ACM	41
SpringerLink	74
ISI WOK	160
ScienceDirect	71
Wiley Interscience	204
Total	**659**
Total (without duplicates)	**607**

Table 3: Search strategy execution results.

Table 3 indicates how many studies we collected from each data sources. We found 659 studies in total. Then, we performed a duplicates check and we found 52 duplicates. Finally, we performed study selection on the remaining 607 candidate studies.

3.1.2 Study selection results

Study selection has been performed using the inclusion and exclusion criteria of Table 2. As the previous stage, we used a spreadsheet to collect data related to the study selection. The spreadsheet includes the following fields:

Selected – Whether the study is selected or not;
Name – the title of the current study;
Year – the publication year of the current study;
I1 to **I4** – Inclusion criteria fulfilling; if it is marked with a ×, the study fulfills the current inclusion criterion, otherwise it does not;
E1 to **E4** – Exclusion criteria fulfilling; if it is marked with a ×, the study fulfills the current exclusion criterion, otherwise it does not;
T *(title)* – If it is marked with a ×, the study has been analyzed reading the title only;
A *(abstract)* – If it is marked with a ×, the study has been analyzed reading the abstract only;
F *(full-text)* – If it is marked with a ×, the study has been analyzed by means of a full-text reading;
Table 4 shows how many studies have been selected or discarded according to the fulfilling of these criteria. In addition, in Table 5 we show how studies have been analyzed with respect to the level of reading detail.
Then, we performed the cross-reference check on the 13 selected primary studies.

3.1.3 Cross-reference check results

Cross-reference check (CRC) has been performed on the References section of each selected study. Every cited content has been collected in a spreadsheet having the same fields of the spreadsheet used in the Study selection stage. We collected 389 new studies.
Then, study selection among references has been performed following the same criteria used during the study selection. We collected 10 further primary studies, as Table 6 shows.

Criteria	# of Studies
I1 & I2 & I3 & I4	**13**
E1	515
E2.1	29
E2.2	0
E3	43
E4	7

Table 4: Study selection results.

Reading detail	# of Studies
Title	315
Abstract	231
Full-text	61

Table 5: Study selection reading detail.

Criteria	# of Studies
I1 & I2 & I3 & I4	**10**
E1	138
E2.1	38
E2.2	12
E3	141
E4	0

Table 6: CRC results.

Reading detail	# of Studies
Title	110
Abstract	128
Full-text	97

Table 7: CRC reading detail.

In addition, in Table 7 we show how studies have been analyzed with respect to the level of reading detail.
In this stage, we found 6 studies already collected in study selection stage and, hence, they have been discarded without being analyzed with respect to inclusion and exclusion criteria. In addition, 30 referenced studies appear more than once and, hence, they have been labelled as duplicates. Furthermore, 14 studies have been discarded because no related literature resource was available.
Finally, we performed data extraction and synthesis on the 23 selected primary studies, resulting both from study selection and CRC stages.

3.2 Found metrics

In this section, we elicit both SR and NSR metrics, counting how many metrics have been claimed in more than one study and providing a brief description for each metrics.
Appendix A and Appendix B show how many different metrics we found in the selected primary studies. We found 130 metrics, in 23 studies, from which we identified 95 different

Metrics Type	Total	Measurement Unit(s)
Energy	48	Joule (J), Index, Watt (W), Ampere (A) Kilowatt-hour (kWh), Number, byte/kWh
Performance	19	GFLOPS/kWh, Computing Unit/kWh, Percentage (%), Seconds (s), Index, Number
Utilization	17	Percentage (%), Megabyte (MB), Megahertz (MHz), GB/s
Economic	9	Dollars ($)
Performance / Energy	2	GFLOPS/Watt, Index
Pollution	1	CO_2 units

Table 8: Found metrics designed for measuring software energy consumption

metrics.
Among these metrics, 66 are relevant for measuring software energy consumption (SR metrics) and they are listed and briefly described in Appendix A. Remaining 29 metrics, elicited and described in Appendix B, are not directly defined to measure the software energy consumption (NSR metrics), although they are discussed within studies related to green software metrics.
Since we are interested in metrics that measure the energy consumption of software, hereafter we deal with SR metrics only.

3.3 Metric types

In this subsection, we discuss about the type of extracted SR metrics. For each SR metrics, we provide a description, an exemplifying metric of that type, and the related measurement unit(s) used to express results.
Table 8 shows how many metrics have been classified by means of a certain type – which is the kind of results generated by the measurement – with the related measurement unit. In this section, we provide a detailed description of each type listed in Table 8. Moreover, we explain how measurement results are expressed describing measurement units related to each Metrics Type.
Energy type defines metrics designed to measure power and energy consumption (or saving). An Energy metric can be used to perform evaluations on software components, as well as it can be used to estimate software energy consumption at the architectural level. For example, the *Application Performance* metric, claimed by Kipp et al. in [18], is classified as Energy type because it measures energy consumption per computing application unit.
The results of measurements performed by metrics of this type can be expressed by means of several units.
Energy type measurement results can be expressed by means of the following units:

- **Joule** (J), which is a derived unit of energy, according to the SI[10];
- **Index**, which is a generic measurement unit that can be defined as follows:
  - A metric is explicitly given as an index, e.g. the *Workload* metric claimed by Kipp et al. in both [17] and [18];
  - A metric depends on two or more dimensions to be evaluated, e.g. the *Execution Plan Energy Efficiency* metric proposed by Ferreira et al. in [26], which depends on the execution time dimension (expressed in Seconds) and the energy consumption dimension (expressed in Watt-hour – Wh – or KiloWatt-hour – KWh).
- **Watt** (W), which is a unit that indicates the power that in one second gives rise to energy of 1 joule, according to the SI;
- **Ampere** (A) that is the unit of electric current;
- **Kilowatt-hour** (kWh), which is a unit of energy equivalent to one kilowatt of power expended for one hour of time;
- **Number**, which is a unit that enumerates the items under validation;
- **byte/KWh**, which expresses the ratio between of work output and the consumed electric energy.

Performance type defines metrics proposed to measure performance indices, e.g. throughput, response time. To show a valid example, we can refer to *Throughput* metric, claimed by Kipp et al. in [18]. Throughput metric is classified as Performance type because it measure a performance index, that is The number of service requests served at a given time period.
Performance metrics express results in terms of the following units:

- **GFLOPS/KWh** that measures the ratio between the computing performance (GFLOPS, GigaFLOPS or 10^9 FLOPS) and the related power consumed in a period of time (KWh);
- **Computing Unit/KWh** unit, which represents the energy consumption generated by each computing application unit;
- **Percentage** (%), which may be a rate, number, or amount expressed in each hundred;
- **Seconds** (s) that is the unit of time, according to the SI;
- **Index** (see description above);
- **Number** (see description above).

[10]International System of Units, SI, provided by the Bureau International des Poids et Mesures. http://www.bipm.org/utils/common/pdf/si_brochure_8_en.pdf

Utilization type is related to measurement of computing resources, such as hard disk, storage, memory, and I/O operations. For instance, the *CPU Usage* metric – proposed by Kipp et al. in [17] – is Utilization type metric because deals with the relative CPU utilization of specific applications. This type of metrics expresses results in terms of:

- **Percentage** (%) (see description above);
- **Megabyte** (MB) that is a multiple (1024^2) of a byte, which is a unit of digital information in computing and telecommunications;
- **Megahertz** (MHz) that is a multiple (10^6) of Hertz, a unit commonly used to quantify the clock rate of central processing units. It is associated to metrics of IT Resource type;
- **GB/s** that indicates the ratio between Gigabytes (GB) processed and the time elapsed expressed in seconds (s).

Economic type metrics are dedicated to evaluate or to estimate the costs of green policies application or, more simply, the costs of software development, at any stage. Furthermore, Economic metrics measure the receipts earned from energy savings. For instance, the *Lifecycle Cost* metric from [17] is classified Economic type because it measures the total process lifecycle expenditures, including costs of conceptual modeling, analysis, design, development, deployment, maintenance, and evolution.
Economic metrics express results in terms of **Dollars** ($), which is the basic monetary unit of the US and it is used as a common currency to express economic values.

Performance/Energy is a hybrid type and defines metrics that measure both performance dimension and energy consumption (or saving). For example, Chen et al. propose in [8] the *Power Efficiency* metric. It is classified as Performance/Energy because it evaluates how efficiently the power is used within a system; it is a measurement of energy consumption with respect to the service throughput (successfully processed operations per second).
Results from this type of metrics are expressed in terms of the following units:

- **GFLOPS/Watt**, which indicates the ratio between the computing performance (GFLOPS, GigaFLOPS or 10^9 FLOPS) and the power consumed (Watt);
- **Index** (see description above).

Pollution type defines metrics related to the measurement of pollution generated by software usage or development. For example, the *Supply Chain* metric claimed by Kipp et al. in [17] is Pollution type metric because it defines the index of carbon emissions, being caused by transportation, logistics, etc. needed for the execution of services.
Pollution metrics results are expressed in **CO_2 units**, which indicate the amount of pollution generated by the execution of services or applications.

3.4 Metric contexts

In this subsection, we describe in detail the metrics context we identified during the data synthesis stage. For each metrics context, we provide a description and an example of metric that belongs to that context.

Metrics Context	Total
Application	35
Architecture	17
Service	13
Service Center	11
Virtual Machine	10
Data Center	3
Embedded Software	3
Server	3
DBMS	1

Table 9: Contexts to classify environment in which metrics are involved.

Table 9 shows how many metrics belong to a certain context. In this section, we provide a detailed description of each of those contexts.
Application context defines metrics that deal with software programs or pieces of code. Application context also involve those metrics that perform measurements related to the application lifecycle, from the design to the maintenance stage. An example of metric belonging to the Application context is the *Computational Energy Cost* metric – claimed by Sharma et al. in [33] –, since it evaluates the energy cost due to CPU processing, memory access, I/O operations of an application.
Architecture context groups metrics that are designed to estimate energy consumption at the design stage. For example, the *Distributed System Energy Consumption* metric, claimed in [31] by Seo et al., belongs to the Architecture context because it estimates the energy consumed by a distributed system as the sum of the energy consumed by its constituent components and connectors, which are architectural elements.
Service context is related to metrics that measure energy consumption generated by the execution and the development of software service. For instance, the *Execution Plan Energy Efficiency* proposed by Ferreira et al. in [26] is a metric belonging to the Service context because it measures how efficiently a service uses energy.
Service Center context contains all those metrics that measure the impact of service execution on a service center. A valid example of metric belonging to the Service Center context is the *Service Center Energy Consumption*, claimed by Cioara et al. in [9]. This metric is belonging to the Service Center context because it measures the energy consumption generated by both active servers – which are executing services – and idle servers that are part of a service center.
Virtual Machine context groups metrics that are designed to estimate or to evaluate energy consumption generated by virtual machines. Furthermore, this context includes metrics that measure the effects of virtualization on energy consumption. For instance, the *Disk Energy Model* proposed in [16] by Kansal et al. belongs to the Virtual Machine context because it represents the energy consumed by the disk over time duration for a single virtual machine.

Data Center context contains all those metrics that perform measurements of the impact of data storing and retrieval on a data center. For instance, the *Data Centre Energy Productivity (DCeP)* metric introduced by Laszewski and Wang in [35] belongs to the Data Center context because it measures the number of bytes that are processed (useful work) per kWh of electric energy with respect to the whole data center.
Embedded Software context contains metrics that are used to perform evaluations or estimations on software that interacts directly with the physical world, e.g. the *Executed Instructions Count Measure (EIC)*, claimed in [7] by Chatzigeorgiou and Stephanides, belongs to this context because it evaluates the energy consumption dealing with the number of executed assembly instructions and considering a typical embedded integer processor core.
Server context includes metrics that perform measurements on the impact of application, service or data processing on a server machine. Although this context may enclose both the Service Center and the Data Center contexts, we specify the Server context as a separate context in order to classify those metrics that have been claimed without any precise definition of their environment, but that are related to servers. A valid example of a metric fitting with the Server context is the *Server Power Utilization* metric, claimed by Gmach et al. in [13], which evaluates the amount of power used by a server with respect to the server CPU utilization.
DBMS context describes all those metrics that are designed to measure the energy cost of data storing and retrieving operations. Metrics belonging on this context are mainly focused on the calculation of energy consumption generated by queries and they can be used for optimize query structures, in order to guarantee energy savings. For example, the *Aggregated Cost* metric proposed by Xu in [24] belongs to the DBMS context because it evaluates a power-aware query plan, in order to improve the energy efficiency of the executed queries.

3.5 Measured resources

In this seaction we describe in detail the resources on which measurements are performed by means of SR metrics. For each resource, we provide a brief textual description of the resource, the way the resource is measured, and an example that shows how a metric measures a resource.

Connectors and components of software architecture define **Architectural Elements** resource. They are measured in several ways, according to the related architectural style, in order to perform energy consumption estimation. For example, the *Remote Client Energy Cost* metric – claimed by Seo et al. in [31] – estimates the energy consumption of a client connector due to sending requests and receiving responses.
Application resource is a program or piece of software designed and written to fulfill a particular purpose of the user. It is measured in terms of workload, runtime platform configuration, request type and rate, data exchange, and power consumption. For example, the *First Order Software Energy Estimation Model*, proposed by Sinha and Chandrakasan in [34], measures the amount of electricity consumed by a program during its execution.
We define **Service** resource as a set of units of functionality that are unassociated and loosely coupled, have no calls to each other embedded in them, implement one single action.

Measured resource	Total
Architecture Elements	17
Application	13
Service	11
Financial Impact	9
Memory	9
CPU	7
Storage	5
Virtual Machines	5
Source Code	3
Data Center	2
Power	2
Process	2
Service Center	2
Service Execution Path	2
DBMS	1
IT Resource	1
JVM	1
Network	1
Pollution	1
Server	1
System	1

Table 10: Resources measured by SR metrics.

It is measured in terms of performance, e.g. availability, response time, or reliability. For instance, the *Throughput* metric, proposed by Kipp et al. both in [17] and in [18], measures the number of service requests served in a given time period.
We define **Financial Impact** as the expenditures or receipts generated by the adoption of a certain solution. It is measured with respect to: energy consumption or saving, application lifecycle, and regulations compliance. One relevant example for measured expenditures is the *Compliance* metric, claimed by Kipp et al. in [18], which represents the cost of guaranteeing conformity degree about regulations and policies established by third parties. By contrast, a significant instance of income measurement is the *Average Revenue* metric proposed by Mazzucco and Dyachuk in [24], which defines the average profit generated by successfully processed jobs against energy consumption costs.
Memory resource is defined as random-access memory units used to store and retrieve non-permanent data. It is measured as bytes of occupied memory or the percentage of RAM utilization for storing/retrieving operations. For instance, the *Memory Energy Model*, claimed by Kansal et al. in [16], estimates the energy consumed by the memory over a given time duration.
The **CPU** (central processing unit) resource is the responsible of interpretation and execution of program instructions. It is possible to measure its utilization or to evaluate the maximum allowed clock frequency with respect to predefined thresholds. For example, the *Application Server Usage* metric, proposed by Kipp et al. in [19], allows getting an overview over the CPU utilization and the amount of disk I/O operations.
Storage is intended as a nonvolatile memory of large amounts of information in electronic form. It is measured in terms of bytes occupied on storage devices, energy consumption generated by writing and reading operations or percentage

of hard disk utilization. For example, the *Storage Usage* metric, mentioned by Kipp et al. in [17], denotes the whole storage utilization percentages for data-related operations on the corresponding storage device for an application, in a given configuration.

Virtual Machine is a software implementation of a computer that executes programs like a physical machine. It is measured in terms of processed bytes or power consumption and efficiency. For instance, the *Energy Savings* metrics introduced by Dhiman et al. in [11] measures the energy reduction in executing each combination of virtual machines using vGreen – a multi-tiered software system for energy efficient computing in virtualized environments – over E+, an enhanced version of the virtual machines scheduler Eucalyptus [27].

Source Code is a set of software instructions, written in a certain language. It is measured as number of executed instructions, number of memory accesses performed or the average energy consumption of each instruction. An example of Source Code resource measurement is the *Executed Instructions Count Measure (EIC)*, proposed by Chatzigeorgiou and Stephanides in [7], that represents the number of executed assembly instructions considering a typical embedded integer processor core.

A **Data Center** is a centralized repository for the storage, management, and dissemination of data and information. It is measured in terms of energy efficiency against the data center work output. An example of Data Center measurement is the *Data Center Energy Productivity metric (DCEP)*, mentioned by Laszewski and Wang in [35], which evaluates the energy efficiency calculating the number of bytes that are processed per kWh.

We refer to **Power** as the rate at which energy is transferred, used, or transformed. More in detail, it is the amount of energy used by a software component. It is measured in terms of usage and effectiveness. For instance, Kipp et al. introduce in [17] the *System Power Usage* metric, which refers to the power consumption of a system running the application, including the power consumption of the computer system whilst taking into account the according facility and infrastructure energy consumption.

Process resource is defined as the set of software development stages. It is measured in terms of energy consumption generated by the adoption of a certain development style, which is the set of methods and techniques to design, implement and deploy a software system. An example of Process resource measurement is the *Process Engineering* metric, introduced by Kipp et al. in both [17] and [18], which evaluates factors regarding the quality of the adopted platform and the quality of the developed code.

A **Service Center** is an operating software unit within a certain organization that provides a service or a group of services to users. It is measured in terms of energy consumption and performance. For example, the *Service Center Energy Consumption* metric, claimed by Cioara et al. in [9], evaluates the energy consumption of the service center in normal operation.

A **Service Execution Path** is a set of executed tasks, which identifies all possible execution scenarios of a composite service. It is measured in terms of energy consumption and efficiency. For example, the *Energy Consumption* metric, mentioned by Ferreira et al. in [26], is the measure of the total energy consumed by a service during its execution, with respect to a concrete execution plan.

A **Computing Unit** is a portion of an application or a system that processes a certain type of data. It is measured in terms of energy consumption. For instance, the energy consumption of each computing application unit is evaluated by the *Application Performance* metric, which is introduced by Kipp et al. in [18].

A **DBMS** is a software package with computer programs that control the creation and the usage of a set of databases. It is measured in terms of energy cost of every single query and used to create an optimized version of queries that require too much energy, with respect to a predefined threshold. For example, the *Aggregated Cost* metric, claimed by Xu in [37], evaluates the superiority of a power-aware query plan, in terms of power and energy costs.

IT Resource is intended as a generic IT resource, such as CPU, memory, and storage. It is worth to note that this kind of resource is used only if no specific IT resource is mentioned. It is measured in terms of energy consumption and efficiency. An example of IT Resource measurement is the *Asset Efficiency* metric mentioned by Kipp et al. in [18], which evaluates the efficiency off all system IT resources in terms of energy and utilization.

A **JVM** (Java Virtual Machine) is a virtual machine that executes Java bytecode. It is measured in terms of energy overhead. For instance, the *Infrastructure Energy Overhead* metric, proposed by Sharma et al. in [33], evaluates the energy costs for executing a Java component incurred by JVM's garbage collection and implicit OS routines, aggregating the energy costs of all the components and the infrastructure energy overhead of all JVMs.

Network resource is defined as a number of interconnected computers, machines, or operations. It is measured in terms of data exchanged over the network itself. For example, in [33] Sharma et al. introduce the *Communication Energy Cost*, which measures the energy cost due to the data exchanged over the network.

Pollution is the "presence in or introduction into the environment of a substance, which has harmful or poisonous effects[11]". It is measured in terms of the amount of carbon emissions due to execution of a service. For example, the *Supply Chain* metric introduced by Kipp et al. in [17] measures the carbon emissions, being caused by transportation, logistics, etc. needed for the execution of the according services.

A **Server** is a computer or computer program that manages access to a centralized resource or a service in a network. It is measured in terms of power consumption. For instance, Gmach et al. in [13] propose the *Server Power Utilization*, which evaluates the amount of power used by a server with respect to its CPU utilization.

System resource is intended as a set of interacting or interdependent software components forming an integrated whole. It is worth to note that the term "system" is occasionally used as synonym of application. In those cases, the metrics have been associated to the Application resource. System is measured in terms of power consumption. As an example of System resource measurement we can refer to the *System Energy Model*, claimed by Kansal et al. in [16], that denotes the full system power consumption.

[11] http://oxforddictionaries.com/definition/pollution

3.6 Metric purposes

In this section, we show how many metrics are used to measure or estimate software energy consumption. Table 11

Metrics usage	Total
Measurement	63
Estimation	29
Both	4

Table 11: Amount of metrics designed for measurement and/or estimation of software energy consumption.

shows how many metrics have been involved in measurement or estimation.

Measurement is the real assessment of a dimension that can be performed by a metric or the calculation of a value based on one or more measurements. For example, the Server Power Utilization metric proposed by Gmach et al. in [13] assesses the amount of power used by a server with respect to its CPU utilization. By constrast, **estimation** is a theoretical prediction or approximation on future values of a dimension executed by a metric. For instance, the *Distributed System Energy Consumption* metric, claimed by Seo et al. in [31], that estimates the energy consumed by a distributed system at the architectural level modeling it as the sum of the estimated energy consumed by its constituent components and connectors.

Surprisingly, most of the extracted metrics are designed for measurement purpose, whilst only 30 metrics are meant to estimate energy consumption. Furthermore, it is worth to mention that 17 out of 30 estimation metrics are claimed in a single study [31]. The majority of metrics related to the measurement purporse should be due to the several kind of results that those metrics are designed for. Indeed, results generated by Measurement metrics are related not only to energy consmption, but also to the financial effects of energy consumption and saving, to the utilization of IT resources that are responsible of energy consumption, and to the effects of energy consumption and savings on the system performance.

Finally, only 4 metrics are designed to estimate and measure energy consumption.

4. ANALYSIS

In the previous section, we showed the results of data extraction from the selected primary studies. In this section, we want to make a further step: we analyze extracted data in order to obtain answer to the following issues.

In Section 4.1, we want to show what is the research direction with respect to green software metrics, in terms of the aim, the environment, and the kind of measurement results of those metrics. In Section 4.2, we want to focus on the way resources are measured, in order to show which resource plays a central role in the measurement of software energy consumption. In Section 4.3 we show which type of metrics are more appealing for each context.

4.1 Which is the focus of research in green software metrics in the last decade?

In this section, we want to show the research direction in terms of the environment and the goal of measurements and in terms of usage of those metrics. This is possible focusing our analysis on the metrics context and on the measured resources, in the time range we led our study selection.

Then, we show which are the trends related to the last decade for metrics types, contexts and measured resources, answering to the following questions:

1. Which metric types have been investigated during the last decade?
2. Which metric contexts have been explored during the last decade?
3. Which resources have been measurend in the last decade?

It is worth to mention that figures and tables related to this section contain complete data until 2011, whilst data about collected studies published in 2012 are partial, since the search has been performed in the early 2012.

4.1.1 Which metric types have been investigated during the last decade?

Table 12 shows the trend of metrics type used in the last decade.

Metrics designed to measure energy – **Energy** type metrics – have been proposed since the beginning of our time range, with a notable peak in 2002, as shown in Figure 1. It is worth to note that this peak is due to the large number of metrics for energy estimation at architectural level proposed by Seo et al. in [31].

Surprisingly, **Economic** type metrics have been claimed only in the last years. As we stated before, energy measurement has been explored since the early 2000s, and this is because IT organizations have been always dealing with energy consumption, since it is one of the common sources of expenses, e.g. energy consumption in service centers, data centers, server farm, and so on. Furthermore, IT organization are interested in measuring energy in order to quantify consumption and to understand how possible is to save energy, in order to reduce energy costs. This is the reason why it is unexpected that metrics designed to measure cost related to energy consumption and saving have been claimed only in the latest year of the 2000s.

In the latest years, research focused on both **Utilization** and **Performance** metrics and the trend shows that the number of metrics of these types is increasing.

Indeed, several studies state that the resources that mostly affect software energy consumption are IT resources such as CPU, memory and storage. For this reason, the number of Utilization metrics is increasing in the last part of the decade. Furthermore, saving energy implies performance issues – i.e. system availability, throughput, and response time. Decreasing energy consumption may negatively affect system performance. Hence, it is worth to measure and monitor system performance in order to balance it with energy saving. For this reason, the number of metrics designed to measure performance – Performance type metrics – has recently increased.

Contexts such as **Pollution** or **Performance / Energy** have been investigated by too few to be significant and, hence, the number of metrics claimed does not affect the general trend of research interest.

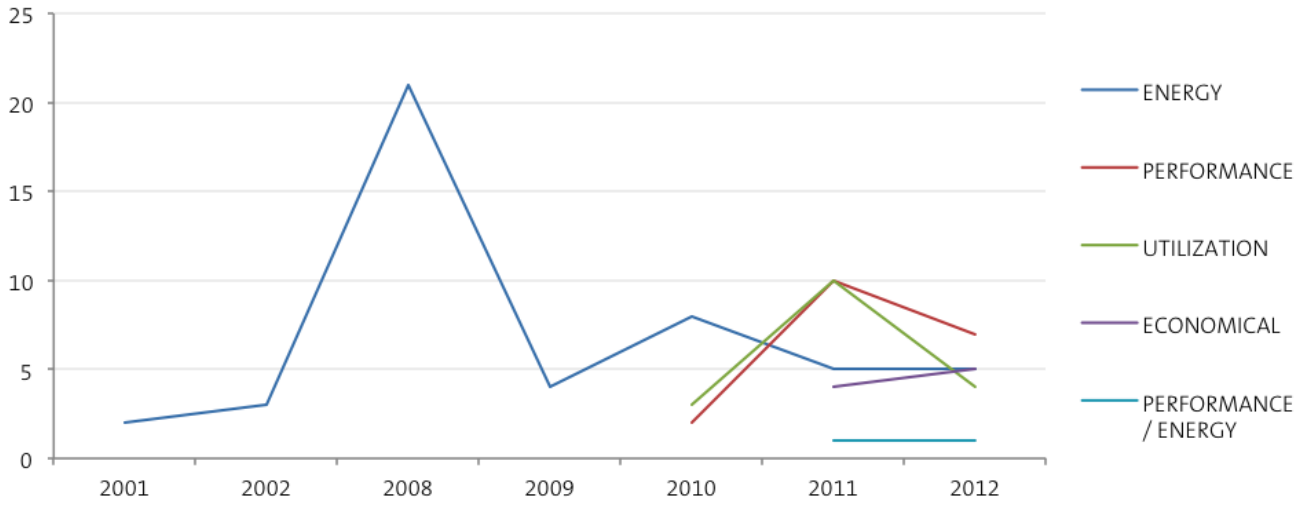

Figure 1: Trend of metrics types from 2001 to 2012.

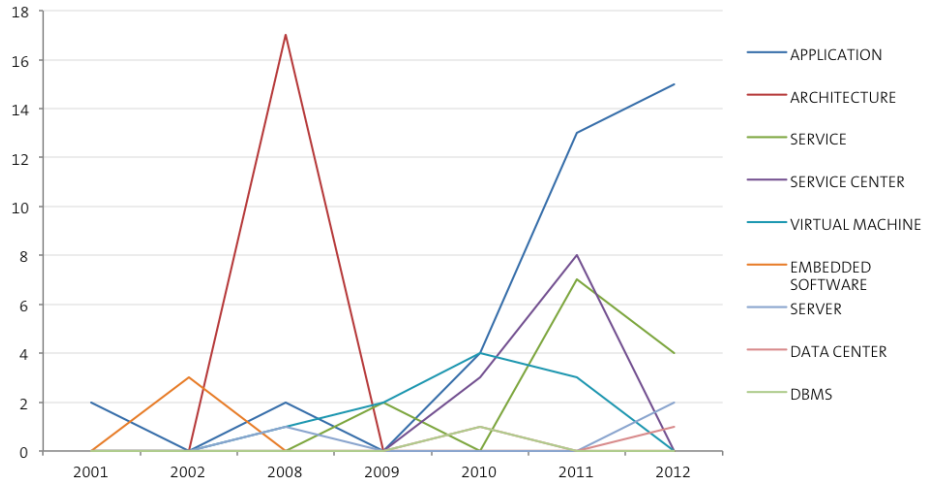

Figure 2: Trend of metrics contexts from 2001 to 2012.

	Years							
Metrics Type	**2001**	**2002**	**2008**	**2009**	**2010**	**2011**	**2012**	**Total**
Energy	2	3	21	4	8	5	5	48
Performance	0	0	0	0	2	10	7	19
Utilization	0	0	0	0	3	10	4	17
Economic	0	0	0	0	0	4	5	9
Performance / Energy	0	0	0	0	0	1	1	2
Pollution	0	0	0	0	0	1	0	1

Table 12: Number of metrics, claimed during the last decade, sorted by type.

4.1.2 *Which metric contexts have been explored during the last decade?*

Table 13 shows how metrics are distributed among contexts and how contexts have been investigated during the last decade.

The **Application** context is the one containing the highest number of metrics and the research interest about this context is keen since the beginning of the 2000s. With the growing of attention about green computing topics in the latest years, the Application context has been explored more in detail and the most of the metrics have been claimed between 2011 and 2012, in several studies.

Different from the Application context, the **Architecture** context has been investigated, from the green metrics perspective, only in one study. Indeed, Seo et al. in [31] propose a set of 17 metrics to estimate energy consumption at the architectural level. Nevertheless, any other study proposed a set of metrics for measuring or estimating energy consumption at the architectural level neither in the recent years nor in the early years of the last decade.

Service context has gained attention only in the last two years. The recent increasing interest in service-oriented development techniques and methods has led the Software Engineering community to focus on energy consumption in the development, implementation and execution of software services.

The **Service Center** context has been explored almost together with the Service context. Indeed, service centers played a central role in deployment of software service as service-oriented systems became more popular and the diffusion of service-oriented systems caused the increase of the number of service centers. For this reason, the number of metrics proposals in the Service Center context grows almost in the same way as the number of metrics claimed for the Service context.

Furthermore, the **Virtual Machines** context has been investigated only recently, from 2008 to 2012. The recent diffusion of cloud computing and high-performance computing (HPC) systems led to the prolixity of virtualized systems, on which applications, services and data are stored and executed. For this reason, energy consumption measurement in virtualized environments became more important recently than before.

Surprisingly, context related to data management and storage – i.e. **DBMS** and **Data Center** contexts – have not been investigated in detail. Cloud-based systems and service-oriented systems deal with a huge amount of data and, hence, storage and withdrawal operations from databases should be performed very frequently. However, only Xu in [37] focuses on the power awareness of DBMS query optimization.

Contexts such as **Embedded Software** or **Server** have been explored by single studies and the number of metrics claimed does not affect the general trend of research interest. Moreover, these contexts rely more on hardware energy consumption than on software. Hence, it is quite expectable to find a few numbers of metrics related to these contexts.

4.1.3 *Which resources have been measurend in the last decade?*

Table 14 shows which resources have been measured with metrics claimed in the last decade.

For sake of clarity and due to the high number of extracted measured resources, we do not show the trend by means of graphic.

Architectural Elements seem to be the resource that is most measured but, as we stated in the previous subsection, metrics involved in this measurement are related to a single study (Seo et al. in [31]). Indeed, architectural elements have not been measured by any other study in the last decade, as depicted in Figure 2.

Application is the resource that has been analyzed for all the duration of the time range we selected. Several studies focused on measuring the energy consumption produced by application developing, implementation and execution.

The constant number of studies and related metrics dedicated to the Application resource means that research is continuously interested in developing and proposing measurement techniques to measure or estimate application energy consumption. Indeed, Figure 2 shows how the number of claimed metrics about measurement performed on Application resource is recently increasing.

Service resource earned more interest during the recent years because of the diffusion of service-oriented design and development techniques and, hence, the prolixity of service-oriented software systems. Services have been measured in terms of energy consumption among their development, deployment and execution.

The economic impact is another important dimension. Indeed, the **Financial Impact** resource has been measured in terms of average revenue from successfully processed jobs [24], expenses for guaranteeing regulations and policies compliance and expenditures related to organizational and application lifecycle factors [17] [18].

Finally, several IT resources, such as **Memory**, **CPU** and **Storage**, have been measured and the attention on this kind of resources has been higher in the latest years. As we stated in the Subsection 4.1.1, recent studies stated that IT resources are the principal source of energy consumption within IT system. For this reason, Memory, CPU and Storage have been measured more recently than in the past.

The other resources do not affect the trend of measured resources, since they are discussed only in single studies, e.g. **Pollution**. However, they still do not contribute to analyze and describe the measured resources that are more investigated by the research community.

Once we answered to these questions, we can analyze what is the trend that research about green software metrics has been focusing on in the last decade.

The research community is particularly interested in measuring energy dimensions – both for saving and for consumption –, taking into account how IT resources are used and how much energy may affect system performance. These measures are mostly performed on application and services, focusing especially on their development and execution.

Finally, interest about economic impacts is increasing, in order to quantify expenditures related to energy consumption and revenues related to energy saving.

4.2 How resources have been measured?

In this section, we want to show which kind of results are generated by measurement performed on the resources we elicited in Section 3.5.

First, we defined and analyzed the *type coverage* for measured resources (TC_{mr}), which is the index representing how

	Years							
Metrics Context	**2001**	**2002**	**2008**	**2009**	**2010**	**2011**	**2012**	**Total**
Application	2	0	2	0	4	13	15	36
Architecture	0	0	17	0	0	0	0	17
Service	0	0	0	2	0	7	4	13
Service Center	0	0	0	0	3	8	0	11
Virtual Machine	0	0	1	2	4	3	0	10
Embedded Software	0	3	0	0	0	0	0	3
Server	0	0	1	0	0	0	2	3
Data Center	0	0	0	0	1	0	1	2
Dbms	0	0	0	0	1	0	0	1

Table 13: Number of metrics claimed from 2000 to 2012, sorted by context.

	Years							
Resources	**2001**	**2002**	**2008**	**2009**	**2010**	**2011**	**2012**	**Total**
Architecture Elements	0	17	0	0	0	0	0	17
Application	2	0	1	0	4	2	4	13
Service	0	0	0	0	0	6	5	11
Financial Impact	0	0	0	0	0	4	5	9
Memory	0	0	0	0	2	5	2	9
CPU	0	0	0	0	2	4	1	7
Storage	0	0	0	0	2	2	1	5
Virtual Machines	0	0	0	2	0	3	0	5
Source Code	0	3	0	0	0	0	0	3
Data Center	0	0	0	0	1	0	1	2
Power	0	0	0	0	0	1	1	2
Process	0	0	0	0	0	1	1	2
Service Center	0	0	0	0	0	2	0	2
Service Execution Path	0	0	0	2	0	0	0	2
DBMS	0	0	0	0	1	0	0	1
IT Resource	0	0	0	0	0	0	1	1
JVM	0	0	1	0	0	0	0	1
Network	0	0	1	0	0	0	0	1
Pollution	0	0	0	0	0	1	0	1
Server	0	0	1	0	0	0	0	1
System	0	0	0	0	1	0	0	1

Table 14: Number of resources measured by metric claimed in the last decade.

	Measured Resources																						
Types	Application	Architecture Elements	Financial Impact	CPU	Data Center	DBMS	IT Resource	JVM	Memory	Network	Pollution	Power	Process	Server	Service	Service Center	Service Execution Path	Source Code	Storage	System	Virtual Machines	*MRT*	TC_{mr} (%)
Energy	9	17	0	1	1	0	0	1	1	1	0	2	2	1	0	1	2	3	1	1	4	16	76
Performance	4	0	0	1	1	1	0	0	0	0	0	0	0	0	11	1	0	0	0	0	0	6	28
Utilization	0	0	0	5	0	0	0	0	8	0	0	0	0	0	0	0	0	0	4	0	0	3	14
Performance / Energy	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	9
Economic	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4
Pollution	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	4

Table 15: Relation between metrics type and related measured resources.

resources have been measured (mr) by a metric of certain type. Let MRT be the number of measured resources associated to a certain type and R the total number of measured resources, TC_{mr} is calculated as follows:

$$TC_{mr} = \frac{MRT}{R} * 100$$

Then, we show which resources can be measured in different ways, by metrics of different type.

Energy type metrics are used to measure the majority of the elicited resources. Indeed, 16 out of 25 extracted resources have been measured by metrics of type Energy. This value shows that energy consumption and energy saving can be measured from several perspectives.

Utilization type metrics are used - by definition - to measure resources such as memory, CPU and storage. Indeed, 18 metrics of Utilization type are distributed among these three resources, with the exception of the *Application Performance* metric, proposed by Kipp et al. in [19], that provides a value for comparison of different IT service centers against performance values according to the energy consumption and performs the measurement on the whole Service Center resource.

Metrics of **Performance** type are mainly focused on measuring performance dimensions related to services. However, some Performance metrics are also designed to measure IT resources, such as CPU, and the performance of a system or an application.

As expected, **Economic** metrics are designed to measure costs, terms of revenue from, expenses for regulations compliance and organization expenditures.

Examining the type coverage of the mentioned resources, we figured out the some resources can be measured by metrics of different type.

For example CPU is measured by 5 Utilization metrics, 1 Energy metric and 1 Performance metric. In case of Utilization, CPU is measured in terms of the percentage of time that the allocated CPU spends for processing the instructions of the applications. Regarding to Energy type, the metric estimates the energy consumption of the CPU with respect to its utilization. In respect of Performance type, CPU and disk I/O operation are measured together, getting an overview over the CPU utilization and the amount of Disk I/O operations. Memory resource is a further example of resource that is measured in different way. Although it is mainly measured by IT Resource metrics, Memory is measured to estimate the energy used by the memory itself during writing and reading operations.

We depicted the overlap among types and measured resources in Figure 3.

To create this figure, we sorted the Metric Types with respect to their TC_{mr} value, and then we selected those measured resources that have been measured by more than one Metric Type. Hence, we created the set of measured resources for each selected Metric Type and we showed the intersections among these sets, confirming that IT resources such as CPU and Memory are the resources that are measured by the most relevant types. Furthermore, we proved that those resources are measured to obtain values in terms of energy consumption and performance.

Finally, we can state that resources are mostly measured to evaluate energy consumption (or saving), to assess energy effects on performance and to monitor IT resources utilization.

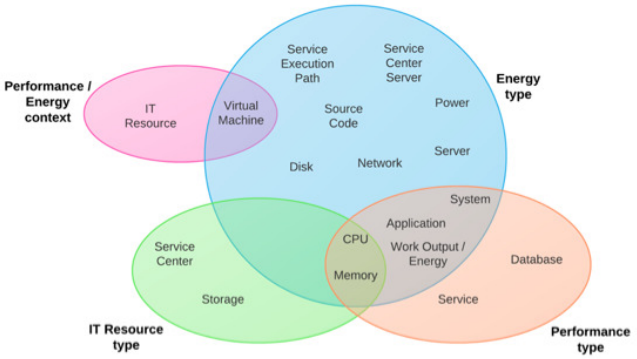

Figure 3: Development process after separation between Metrics Group and Development Group.

4.3 Which types are more appealing for each context?

In this section, we want to show which kinds of result are more interesting with respect to the context which the metrics belongs to. In other words, we want to illustrate what kind of metrics is more appropriate regarding to the environment in which they are involved.

	Contexts										
Types	Application	Architecture	Data Center	DBMS	Embedded Software	Server	Service	Service Center	Virtual Machine	CT	TC_c (%)
Energy	13	17	2	0	3	1	2	1	9	8	89
Performance	5	0	1	1	0	0	10	2	0	5	56
Economic	7	0	0	0	0	2	0	0	0	2	22
Performance / Energy	0	0	1	0	0	0	0	0	1	2	22
Utilization	9	0	0	0	0	0	0	8	0	2	22
Pollution	0	0	0	0	0	0	1	0	0	1	11

Table 16: Relation between metrics types and contexts.

First, we defined and analyzed the *type coverage* for contexts (TC_c), which is the index representing how many contexts (c) have been explored by metrics of certain type. Let CT be the number of contexts associated to a certain type and C the total number of contexts, TC_c is calculated as follows:

$$TC_c = \frac{CT}{C} * 100$$

As shown in Figure 4, **Energy** type is associated to every contexts but DBMS. In each of them, energy consumption is one of the most important dimension to be measured.

Although DBMS context has no Energy type metrics defined, the *Aggregated Cost* metric – proposed by Xu in [24] and classified as Performance type – takes into account the power reduction as an energy factor in the calculation of performance loss.

Performance type metrics are focused on context in which the performance evaluation is relevant, especially for measuring performance effects of energy reduction. Unexpectedly, **Economic** type metrics are claimed only in Application and Server contexts. It is legitimate to ask why the financial impact is not measured in contexts such as Data Center and Service Center. Data centers deal with a relevant amount of data and, hence, with a high number of storing and retrieving operations. Then, data centers require a considerable amount of energy to perform those operations and guarantee availability of data systems. For this reason, it should be interesting to measure the financial impact of energy consumption in the Data Center context, in order to quantify the related energy cost and eventualy search for a solution to optimize the energy consumption.

As for data centers, service centers should guarantee availability of provided services and, for this reason, IT organizations that are providing this services have to deal with energy consumption costs. Furthermore, IT organizations tend to save energy in order to reduce costs. Therefore, Economic type metrics should be defined within Service Center context, in order to measure the economic influence on service-related energy consumption.

Another surprising data that stands out from this analysis is related to the **Utilization** type. Metrics of this type have been claimed only for Application and Service Center contexts.

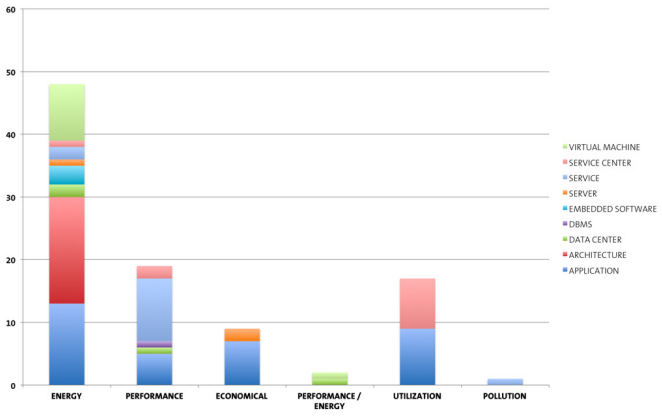

Figure 4: Development process after separation between Metrics Group and Development Group.

We think that Utilization type metrics should be claimed for Data Center and Virtual Machine context as well. Data centers use a notable amount of storage devices that generate a substantial consumption of energy. Therefore, analyzing the utilization of this devices in terms of stored and retrieved data should help to evaluate the energy consumption related to these operations. For this reason, Utilization metrics for Data Center context should be defined.

Furthermore, the use of virtual machines make the computational costs increase, since several virtual machines can be installed on a single server. Therefore, monitoring and evaluating the utilization of CPU in the Virtual Machine context should help to measure the energy consumption related to this IT resource. For this reason, we state that Utilization type metrics for Virtual Machine context should be specified.

5. CONCLUSIONS

In this work, we showed how many green software metrics are claimed in the software engineering literature. Furthermore, we showed how these metrics can be classified in terms of measured resources, kind of results (type), environment or application domain (context), and usage.

Moreover, we verified that research community is focusing on metrics strictly related to energy consumption and saving dimensions. In addition, we found out that claimed metrics show how IT resources utilization is relevant for measuring energy consumption and how much energy may affect system performance. Furthermore, we showed which resources are more measured and how the metrics calculate their energy consumption.

In addition, we showed which metric types are more attractive for defined contexts, discussing which metrics are expected to be defined for contexts that are not explored.

The metrics classification of this work is useful to quickly access to the most important software energy consumption metrics claimed in the software engineering literature. Furthermore, the results of this study represent a starting point for those that are interested in creating models or performing measurements about energy consumption of software. Indeed, these results allow the readers to select accurately the right set of metrics that is more appropriate to their needs, in terms of context, type and resources to be measured.

6. REFERENCES

[1] F. Albertao, J. Xiao, C. Tian, Y. Lu, K. Q. Zhang, and C. Liu. Measuring the Sustainability Performance of Software Projects. In *Proceedings of 2010 IEEE 7th International Conference on e-Business Engineering*, 2010.

[2] N. Amsel, Z. Ibrahim, A. Malik, and B. Tomlinson. Toward sustainable software engineering (NIER track). In *Proceedings of the 33rd International Conference on Software Engineering*, ICSE '11, pages 976–979, New York, NY, USA, 2011. ACM.

[3] I. Anghel, T. Cioara, I. Salomie, G. Copil, and D. Moldovan. An autonomic algorithm for energy efficiency in service centers. In *Proceedings of the Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing*, ICCP '10, pages 281–288, Washington, DC, USA, 2010. IEEE Computer Society.

[4] B. S. Arnaud and D. MacLean. ICTs, Innovation and the Challenge of Climate Change. *IISD*, 2008.

[5] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.*, 80(4):571–583, Apr. 2007.

[6] E. Capra, C. Francalanci, and S. A. Slaughter. Is software "green"? Application development environments and energy efficiency in open source

applications. *Inf. Softw. Technol.*, 54(1):60–71, Jan. 2012.

[7] A. Chatzigeorgiou and G. Stephanides. Energy Metric for Software Systems. *Software Quality Control*, 10(4):355–371, Dec. 2002.

[8] Q. Chen, P. Grosso, K. v. d. Veldt, C. d. Laat, R. Hofman, and H. Bal. Profiling Energy Consumption of VMs for Green Cloud Computing. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, pages 768 –775, dec. 2011.

[9] T. Cioara, I. Anghel, I. Salomie, G. Copil, D. Moldovan, and B. Pernici. A context aware self-adapting algorithm for managing the energy efficiency of IT service centres. *UbiCC Journal*, 6, 2011.

[10] B. de Rijk. Sustainability Innovation - Challenges of Green Abundance. Whitepaper, 2008.

[11] G. Dhiman, G. Marchetti, and T. Rosing. vGreen: A System for Energy-Efficient Management of Virtual Machines. *ACM Trans. Design Autom. Electr. Syst.*, 16(1):6, 2010.

[12] L. Erdmann, L. Hilty, L. Goodman, and J. Arnfalk. The Future Impact of ICTs on Environmental Sustainability, EUR Number: EUR 21384 EN. 2004.

[13] D. Gmach, J. Rolia, L. Cherkasova, G. Belrose, T. Turicchi, and A. Kemper. An integrated approach to resource pool management: Policies, efficiency and quality metrics. In *DSN*, pages 326–335, 2008.

[14] T. Greenhalgh and R. Peacock. Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources. *BMJ*, 10 2005.

[15] R. Kahn, R. I. Kat, and C. Pratt. Energy-Aware Storage Benchmarks. *ERCIM News*, 2009.

[16] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya. Virtual machine power metering and provisioning. In *Proceedings of the 1st ACM symposium on Cloud computing*, SoCC '10, pages 39–50, New York, NY, USA, 2010. ACM.

[17] A. Kipp, T. Jiang, and M. Fugini. *Green Metrics for energy-aware IT systems*, pages 241–248. Ieee, 2011.

[18] A. Kipp, T. Jiang, M. Fugini, and I. Salomie. Layered Green Performance Indicators. *Future Gener. Comput. Syst.*, 28(2):478–489, Feb. 2012.

[19] A. Kipp, J. Liu, T. Jiang, D. Khabi, Y. Kovalenko, L. Schubert, M. vor dem Berge, and W. Christmann. Approach towards an energy-aware and energy-efficient high performance computing environment. In *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*, pages 493 –499, aug. 2011.

[20] B. Kitchenham. Procedures for performing systematic reviews. Technical report, Keele University and NICTA, 2004.

[21] P. Kurp. Green computing. *Commun. ACM*, 51(10):11–13, Oct. 2008.

[22] P. Lago and T. Jansen. Creating environmental awareness in service oriented software engineering. In *Proceedings of the 2010 international conference on Service-oriented computing*, ICSOC'10, pages 181–186, Berlin, Heidelberg, 2011. Springer-Verlag.

[23] L. Lefèvre and J.-M. Pierson. Energy Savings in ICT and ICT for Energy Savings. *Towards Green ICT*, 2009.

[24] M. Mazzucco and D. Dyachuk. Balancing electricity bill and performance in server farms with setup costs. *Future Generation Computer Systems*, 28(2):415 – 426, 2012.

[25] G. I. Meijer, T. Brunschwiler, S. Paredes, and B. Michel. Using Waste Heat from Data Centres to Minimize Carbon Dioxide Emission. *ERCIM News*, 2009(79), 2009.

[26] A. Mello Ferreira, K. Kritikos, and B. Pernici. Energy-Aware Design of Service-Based Applications. In *Proceedings of the 7th International Joint Conference on Service-Oriented Computing*, ICSOC-ServiceWave '09, pages 99–114, Berlin, Heidelberg, 2009. Springer-Verlag.

[27] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The Eucalyptus open-source cloud-computing system. In *Proceedings of Cloud Computing and Its Applications'08.*, 2008.

[28] C. Öhman. Design For Energy Awareness. *Towards Green ICT*, 2009.

[29] B. Pernici, D. Ardagna, and C. Cappiello. Business Process Design: Towards Service-Based Green Information Systems. In *E-Government Ict Professionalism and Competences Service Science*, IFIP International Federation for Information Processing. Springer Boston, 2008.

[30] K. Potter. Green IT Services as a Catalyst for Cost Optimization. *Gartner RAS Core Research Note G00163344*, 2008.

[31] C. Seo, G. Edwards, D. Popescu, S. Malek, and N. Medvidovic. A framework for estimating the energy consumption induced by a distributed system's architectural style. In *Proceedings of the 8th international workshop on Specification and verification of component-based systems*, SAVCBS '09, pages 27–34, New York, NY, USA, 2009. ACM.

[32] C. Seo, S. Malek, and N. Medvidovic. Component-Level Energy Consumption Estimation for Distributed Java-Based Software Systems. In *Proceedings of the 11th International Symposium on Component-Based Software Engineering*, CBSE '08, pages 97–113, Berlin, Heidelberg, 2008. Springer-Verlag.

[33] C. Seo, S. Malek, and N. Medvidovic. Estimating the Energy Consumption in Pervasive Java-Based Systems. In *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, PERCOM '08, pages 243–247, Washington, DC, USA, 2008. IEEE Computer Society.

[34] A. Sinha and A. P. Chandrakasan. JouleTrack: a web based tool for software energy profiling. In *Proceedings of the 38th annual Design Automation Conference*, DAC '01, pages 220–225, New York, NY, USA, 2001. ACM.

[35] G. von Laszewski and L. Wang. GreenIT Service Level Agreements. *Proceedings of the Service Level*

Agreements in Grids Workshop, 2009.

[36] J. Williams and L. Curtis. Green: The New Computing Coat of Arms? *IT Professional*, 10:12–16, 2008.

[37] Z. Xu. Building a power-aware database management system. In *Proceedings of the Fourth SIGMOD PhD Workshop on Innovative Database Research*, IDAR '10, pages 1–6, New York, NY, USA, 2010. ACM.

[38] R. Yanggratoke, F. Wuhib, and R. Stadler. Gossip-based Resource Allocation for Green Computing in Large Clouds (long version). Technical report, KTH, ACCESS Linnaeus Centre, 2011. A short version of this paper appears in the 7th International Conference on Network and Service Management,Paris, France, 24-28 October, 2011. QC 20110809.

APPENDIX

In this appendix, we present the metrics we found. Appendix A contains all the 66 metrics that are designed to measure energy consuption of software (SR metrics). Appendix B contains all the 29 metrics that are not designed to measure software energy consumption (NSR metrics), although have been claimed in the selected primary studies.
The following tables are organized as follows: column **Ref** contains the references to the primary studies that claim the metric; column **Name** represents the name of the metric; column **Description** includes a brief textual description of the metric; column **Unit** contains the measurement unit that expresses the result generated by the metric (see Section 3.3); column **Measured Resource** represents the resource measured by the metric (see Section 3.5); column **Type** shows the kind of results returned by the metric (see Section 3.3); column **Context** describes the environment in which the metric is involved (see Section 3.4); column **Purpose** contains MEAS if the metric is designed for perform a measurement, ESTI if the metric performs an approximation or a predicition of a certain value, MEAS / ESTI if the metric is designed to perform both measurement and estimation (see Section 3.6).
Since some metrics are claimed in more than one study, they may belong to different contexts, although they are designed for the same purpose. Furthermore, their results may be expressed in different ways, in terms of unit and type. For these reasons, we refer the related primary study in the **Unit**, **Type**, and **Context**, so that it is possible to differentiate these features.

A. SR METRICS

Ref	Name	Description	Unit	Measured Resource	Type	Context	Purpose
[37]	***Aggregated Cost***	Superiority of a power-aware query plan.	Index	DBMS	Performance	DBMS	MEAS
[6]	***Application Energy Efficiency***	Energy efficiency of different applications with the same workload.	Index	Application	Energy	Application	MEAS
[17], [18], [19]	***Application Performance***	Performance MEAS with respect to energy consumption.	Computing Unit/kWh	Application	Performance	Application [17] [18], Service Center [19]	MEAS
[18]	***Asset Efficiency***	IT resources efficiency in terms of energy and utilization.	Index	IT Resource	Performance / Energy	Data Center	MEAS
[17], [18]	***Availability***	Probability that a request is correctly fulfilled within a maximum expected time frame.	Percentage	Service	Performance	Service	MEAS
[24]	***Average Revenue (Multi-tier Service Model)***	Average revenue, earned by the provider per unit time, in a multi-tier service (e.g. Wikipedia).	Dollars	Financial Impact	Economic	Server	MEAS / ESTI
[24]	***Average Revenue (Single-tier Service Model)***	Average revenue, earned by the provider per unit time, in a single-tier service.	Dollars	Financial Impact	Economic	Server	MEAS / ESTI
[31]	***Client Connector Energy Cost***	Energy cost of a client connector incurred by receiving requests from and forwarding responses to clients	Joule	Architecture Elements	Energy	Architecture	ESTI
[31]	***Client Energy Cost***	Energy cost of a client due to sending requests to and receiving responses from a connector.	Joule	Architecture Elements	Energy	Architecture	ESTI
[31]	***Client-Server Facilitation Energy Cost***	Facilitation energy cost of client and server connectors.	Joule	Architecture Elements	Energy	Architecture	ESTI
[31]	***Communication Energy Consumption***	Energy consumption of communication, which includes the cost of exchanging data both locally or remotely.	Joule	Architecture Elements	Energy	Architecture	ESTI
[32], [33]	***Communication Energy Cost***	Energy cost due to the data exchanged over the network	Joule	Network	Energy	Application	MEAS / ESTI
[17], [18]	***Compliance***	Cost of guaranteeing conformity degree about regulations and policies established by third parties.	Dollars	Financial Impact	Economic	Application	MEAS

[31]	***Component Energy Cost***	Energy cost of a component due to exchanging subscriptions, unsubscriptions, and events with pub-sub connectors.	Joule	Architecture Elements	Energy	Architecture	ESTI
[32], [33]	***Computational Energy Cost***	Energy cost due to CPU processing, memory access, I/O operations	Joule	Application	Energy	Application	MEAS / ESTI
[31]	***Connector Conversion Energy Cost***	Conversion energy cost of a server connector	Joule	Architecture Elements	Energy	Architecture	ESTI
[17]	***Consumable***	Volume of consumable generated by the application during its workflow execution.	Dollars	Financial Impact	Economic	Application	MEAS
[31]	***Conversion Pub-Sub Energy Cost***	Energy cost of a pub-sub connector incurred by marshaling and unmarshaling events that are transmitted remotely.	Joule	Architecture Elements	Energy	Architecture	ESTI
[31]	***Coordination Pub-Sub Energy Cost***	Energy cost of a pub-sub connector incurred by performing coordination.	Joule	Architecture Elements	Energy	Architecture	ESTI
[16]	***CPU Energy Model***	CPU energy consumption.	Joule	CPU	Energy	Virtual Machine	ESTI
[3], [19], [17], [18], [2], [9]	***CPU Usage***	Percentage of CPU Utilization.	Percentage.	CPU	Utilization [3] [17] [18] [2] [9], Performance [19]	Application [17] [18] [2], Service Center [3] [19] [9]	MEAS
[18], [35]	***Data Center Energy Productivity (DCeP)***	Number of bytes which are processed per kWh of electric energy.	Index [35], byte/kWh [18]	Data Center	Energy [18], Performance [35]	Data Center	MEAS
[18]	***Data Center Performance Efficiency***	Effectiveness of power utilization for providing a service.	Index	Power	Energy	Data Center	MEAS
[16]	***Disk Energy Model***	Energy consumed by the disk over time.	Joule	Storage	Energy	Virtual Machine	ESTI
[31]	***Distributed System Energy Consumption***	Energy consumed by a distributed system.	Joule	Architecture Elements	Energy	Architecture	ESTI
[26]	***Energy Consumption***	Measure of the total energy consumed by the service during its execution, with respect to a CEP (concrete execution plan).	Kilowatt-hour	Service Execution Path	Energy	Service	MEAS
[11]	***Energy Savings***	Energy reduction in executing each combination of VMs using vGreen over E+.	Percentage	Virtual Machines	Energy	Virtual Machine	ESTI
[7]	***Executed Instruction Count Measure (EIC)***	The number of executed assembly instructions considering a typical embedded integer processor core.	Number	Source Code	Energy	Embedded Software	MEAS
[26]	***Execution Plan Energy Efficiency***	Measure of how efficiently a service uses energy for a CEP (concrete execution plan).	Index	Service Execution Path	Energy	Service	MEAS

[31]	***Facilitation Energy Cost***	Energy cost of a pub-sub connector incurred by managing subscriptions and publications, finding the set of subscriptions that match each published event, and creating connection objects that implement remote communication.	Joule	Architecture Elements	Energy	Architecture	ESTI
[34]	***First Order Software Energy ESTI Model***	Amount of current consumed by a program during its execution	Ampere	Application	Energy	Application	ESTI
[31]	***Generic Component Energy Cost***	Energy cost of a component	Joule	Architecture Elements	Energy	Architecture	ESTI
[31]	***Generic Connector Energy Cost***	Energy cost of a connector	Joule	Architecture Elements	Energy	Architecture	ESTI
[17], [18]	***Human Resources***	Costs of human factors affecting the software lifecycle	Dollars	Financial Impact	Economic	Application	MEAS
[19], [17], [18]	***I/O Usage***	Percentage of occupation of the corresponding I/O device for communications and the number of messages transferred by an application over a set of system components	Percentage [17], [18] GB/s [19]	Memory	Utilization	Application [18], [17] Service Center [19]	MEAS
[32], [33]	***Infrastructure Energy Consumption***	Energy cost incurred by an OS and an application's runtime platform (e.g., JVM) in the process of managing the execution of user-level applications	Joule	Application [32], JVM [33]	Energy	Application [32], Virtual Machine [33]	MEAS / ESTI
[17], [18]	***Lifecycle Cost***	Total process lifecycle expenditures	Dollars	Financial Impact	Economic	Application	MEAS
[31]	***Local Pub-Sub Connector Energy Cost***	Energy cost of a pub-sub connector due to exchanging subscriptions, unsubscriptions, and events with local pub-sub connectors	Joule	Architecture Elements	Energy	Architecture	ESTI
[7]	***Memory Access Count Measure (MAC)***	The number of memory accesses to the data memory	Number	Source Code	Energy	Embedded Software	MEAS
[16]	***Memory Energy Model***	Energy consumed by the memory over time.	Joule	Memory	Energy	Virtual Machine	ESTI
[3], [19], [17], [18], [9]	***Memory Usage***	Percentage of RAM utilization.	Percentage	Memory	Utilization	Application [17] [18], Service Center [3] [19] [9]	MEAS
[8]	***Power Consumption***	Cumulative power consumption with respect to execution time.	Joule or Watt	Virtual Machines Energy Virtual Machine MEAS	Energy	Virtual Machine	MEAS
[8]	***Power Efficiency***	MEAS of how efficient the power is used	GFLOPS / Watt	Virtual Machines	Performance / Energy	Virtual Machine	MEAS
[17], [18]	***Process Engineering***	Factors regarding the quality of the adopted platform and the quality of the developed code.	Index	Process	Energy	Application	MEAS
[17], [18]	***Process Time/Job Duration***	The average time taken by a service S, from the time of invocation to the time of completion, including delay.	Seconds	Service	Performance	Service	MEAS

[31]	***Pub-Sub Connector Energy Cost***	Energy cost of a pub-sub connector incurred by exchanging subscriptions and events with components.	Joule	Architecture Elements	Energy	Architecture	ESTI
[17], [18]	***Recoverability***	The capability of a service to restore the normal execution after a failure within a given period of time	Index	Service	Performance	Application [18], Service [17]	MEAS
[11]	***Reduction in Power Imbalance***	Percentage of power imbalance reduction in the system.	Virtual Machines	Percentage	Energy	Virtual MachineS	ESTI
[17], [18]	***Reliability***	Probability that a service remains operational to deliver the desired function over a specified period of time.	Index	Service	Performance	Service [17], Application [18]	MEAS
[31]	***Remote Client Energy Cost***	Energy cost of a client connector due to sending requests and receiving responses.	Joule	Architecture Elements	Energy	Architecture	ESTI
[31]	***Remote Pub-Sub Connector Energy Cost***	The energy cost of a pub-sub connector caused by sending/receiving subscriptions and events to/from remote connectors.	Joule	Architecture Elements	Energy	Architecture	ESTI
[17], [18]	***Response time***	The time taken by a service to handle user requests.	Seconds	Service	Performance	Service	MEAS
[34]	***Second Order Software Energy ESTI Model***	Amount of current consumed by a program during its execution with different instruction classes	Ampere	Application	Energy	Application	ESTI
[13]	***Server Power Utilization***	Amount of power used by a server with respect to its CPU utilization.	Watt	Server	Energy	Server	MEAS
[9]	***Service Center Energy Consumption***	Energy consumption of the service center in normal operation	Index	Service Center	Energy	Service Center	MEAS
[31]	***Services Energy Cost***	Energy cost of services that a connector may provide	Joule	Architecture Elements	Energy	Architecture	ESTI
[7]	***Software Energy (SEM)***	The average energy cost of an instruction, that of a data memory access (assuming a RAM data memory), and that of an instruction memory access (assuming a ROM instruction memory).	Number	Source Code	Energy	Embedded Software	MEAS
[6]	***Specific Energy***	Energy absorbed by a system running an application executing a given functional workload involving multiple requests compared to the average energy absorbed by applications belonging to the same functional area (i.e. with the same set of functionalities) to execute the same workload	Index	Application	Energy	Application	ESTI
[3], [9], [17], [18]	***Storage Usage***	Percentage of the storage utilization.	Percentage	Storage	Utilization	Application [17] [18], Service Center [3] [9]	MEAS
[17]	***Supply Chain***	Index of carbon emissions, being caused by transportation, logistics, etc. needed for the execution of the according services.	CO2 units	Pollution	Pollution	Service	MEAS
[16]	***System Energy Model***	ESTI of full system power consumption as sum of CPU, memory and storage power consumption ESTI.	Watt	System	Energy	Virtual Machine	ESTI
[17]	***System Power Usage***	Power consumption of a system running the application	Kilowatt-hour	Power	Energy	Application	MEAS

[32]	***System's Overall Energy Consumption***	System energy consumption with respect to infrastructure and overall consumption.	Joule	Application	Energy	Application	ESTI
[17], [18]	***Throughput***	The number of service requests served at a given time period.	Index	Service	Performance	Service	MEAS
[17]	***Workload***	The type and rate of requests sent to the system included with the execution of software packages and in-house application programs.	Index	Application	Performance	Application	MEAS

B. NSR METRICS

Ref	Name	Description	Unit	Measured Resource	Type	Context	Purpose
[1]	***Abstractness***	How much a package can withstand change.	Index	Application	Lifecycle	Application	MEAS
[8]	***Benchmark Execution Elapsed Time***	Time elapsed during benchmarks execution.	Seconds	Virtual Machine	Time	Virtual Machine	MEAS
[18], [9]	***Compliance of Greenness Level***	Compliance of the level of greenness of a service centre with specific energy-saving requirements.	Index	Policies	Compliance	Service Center	MEAS
[3]	***Context Entropy***	Fulfilling of both KPI and GPI context policies.	Index	Policies	Entropy	Service Center	MEAS
[18]	***Corporate Average Data Centre Efficiency (CADE)***	Data centre efficiency across the entire corporate footprint.	Index	Data Center	Performance / Energy	Data Center	MEAS
[1]	***Defect Density***	Number of defects with respect to system size.	Number	Application	Software Defects	Application	MEAS
[1]	***Distance From Main Sequence***	The ability to introduce changes quickly and cost effectively.	Index	Application	Changes Impact	Application	MEAS
[1]	***Effectiveness***	Ratio of tasks accomplished without help w.r.t. the whole amount of tasks.	Index	Application	Performance	Application	MEAS
[18]	***Entropy Contribution***	Entropy contribution of a GPI/KPI policy	Index	Policies	Context	Service Center	MEAS
[1]	***Error Rate***	Ratio of errors with respect to number of tasks	Index	Application	Interaction	Application	MEAS
[26]	***Expected Service Request Fulfilling Time***	The expected duration in time that a service spends to fulfill a service request, for a CEP (concrete execution plan).	Seconds	Service Execution Path	Time	Service	MEAS
[18]	***Facility Efficiency***	Facilities efficiency in terms of energy and utilization.	Index	Data Center	Performance / Energy	Data Center	MEAS
[18]	***GAMES Context Situation Threshold***	GAMES context situation entropy associated threshold	Index	Policies	Context	Service Center	MEAS
[18]	***Green Level***	Score for evaluating and measuring the greenness of a context can be evaluated as a weighted sum of the evaluation with a function fn of the GPIs included in the context policy.	Index	Policies	Context	Service Center	MEAS
[3], [9]	***Humidity***	Humidity treshold that is allowed in the system environment.	Percentage	Environment	Environment	Service Center	MEAS
[1]	***Instability***	Potential impact of changes in a given package.	Index	Application	Changes Impact	Application	MEAS
[1]	***Learnability***	Ratio of how fast a user can learn to use the application w.r.t. the time the user used it	Index	Application	Interaction	Application	MEAS
[3], [9]	***Light***	Light presence in the system environment	On/Off	Environment	Environment	Service Center	MEAS

[1]	***Long-Haul Roundtrips***	Long distance there and back trips	Number	Employees	Organizational	Organization	MEAS
[8]	***Performance***	Overall performance of a given system under test.	GFLOPS / s	Virtual Machine	Performance	Virtual Machine	MEAS
[38]	***Reduction of Power Consumption***	Fraction of machines in the cloud that are freed by the protocol.	Percentage	Machines	Energy	Cloud	MEAS
[1]	***Relative Response Time***	Amount of tasks fulfilling performance goals	Index	Application	Performance	Application	MEAS
[18]	***Return of Green Investments (RoGI)***	Period of time in which the investments made in green solutions recuperate.	-	Time	Time	Organization	MEAS
[24]	***Server Power***	The amount of electricity consumed by n physical server per unit of time	Watt	Server	Energy	Server	MEAS / ESTI
[3], [9]	***Temperature***	Temperature treshold that is allowed in the system environment	Degrees	Environment	Environment	Service Center	MEAS
[1]	***Testing Effectivenes***	Level of effectiveness of a test	Index	Application	Compliance	Application	MEAS
[1]	***Testing Efficiency***	Level of efficiency of a test	Index	Application	Compliance	Application	MEAS
[11]	***Weighted Speedups***	The average speedup of each VM combination with vGreen.	Percentage	Virtual Machine	Performance	Virtual Machine	ESTI
[1]	***Work-From-Home Days***	Percentage of time spent go from/to work and home	Percentage	Employees	Organizational	Organization	MEAS